正規パターン言語の和と共通部分の帰納学習 (計算機科学基礎理論とその応用)

AUTHOR(S):

植村, 仁

CITATION:

植村, 仁. 正規パターン言語の和と共通部分の帰納学習 (計算機科学基礎理論とその応用). 数理解析研究所講究録 2005, 1426: 45-50

ISSUE DATE:

2005-04

URL:

http://hdl.handle.net/2433/47297

数理解析研究所講究録 1426 巻 2005 年 45-50

正規パターン言語の和と共通部分の帰納学習

植村 仁
Jin Uemura
大阪府立大学総合科学部
Department of Mathematics and Information Sciences
Osaka Prefecture University

1 導入

パターンとは，定数記号及び変数からなる有限文字列あり，パターン言語は，各変数へ定数記号からなる文字列を代入して得られる文字列からなる．空列 ε 代入を許す場合，消去可能パターン言語といい，許さない場合，消去不能パターン言語という．消去不能パターン言語の族は，Gold[5] の枠組みで正例から推論可能な言語族として，Angluin[2] によって導入された．一方，消去不能パターン言語の族は，篠原 [16] により Extended パターン言語の名前で導入された．この言語族の推論可能性の問題は，Reidenbach によって，定数記号が 2,3,4 個の場合にこの問題を否定的に解決した [11, 12]．一方，パターンに含まれる変数が全て異なる場合，正規パターンと呼ばれる．篠原 [16] は，消去可能，消去不能 の双方に対して，正例から多項式時間で推論可能であることを示した．また，有村等 [4] は，高々 k 個のパターン言語の和の族に関して，言語の包含関係とパターン集合の構文的包摂関係の等価性を与える Compactness の概念を導入し，Compactness を有する言語和の族を効率的に学習するアルゴリズムを一般的な枠組みで構築した．そして，Compactness が成立する必要十分条件は 消去不能パターン言語については佐藤等 [14] が，消去可能パターン言語については著者等 [22] が与えた．本稿では，消去可能正規パターン言語の和，積，補集合についての学習問題を扱う．消去不能正規パターン言語の和，積，補集合についての学習は佐藤等 [15] が扱っている．

学習対象となる言語族は k-和積パターン言語と呼ばれる，高々 k 個の消去可能正規パターン言語に和演算，積演算を施して得られる言語の族である．これが正例から帰納学習可能であることを示すために，学習対象となる言語の 2 通りの表現を定義する．1 つは和積標準形であり，もう 1 つは消去可能正規パターン言語の有限和である．消去可能パターン言語の言語和の族の Compactness [22] の結果を用い，2 つの k-和積パターン言語の包含関係を決定づける特徴集合の存在を示す．これにより k-和積パターン言語間の包含関係と k-和積パターン間のある構文的関係の間に同値関係が存在することをも示す．これらの結果から，有限証拠集合が具体的に列挙できることを示し，学習可能性を導く．

最後に，高々 k 個の消去可能正規パターン言語に補演算，和演算，積演算を施して得られる言語の族は正例から帰納学習可能ではないことを示す．Kapur[6] の定義した言語族の集積点の概念とその結果を用い，この言語族が学習可能でないことを示す．

2 和積パターン言語

2.1 正規パターン言語

正規パターン Σ を有限個の定数記号からなる集合であるとし，アルファベットと呼ぶ．a, b, c 等の記号で表す．定数記号の有限列を語と呼び，語の集合を言語と呼ぶ．長さ 0 の語を空列と呼

び, ε で表す. X を変数からなる加算無限集合であるとする. x, y, z 等の記号で表す.

* をクリーネ閉包とする. Σ^* は全ての語の集合を表す. Σ^n を長さ n の語全体, $\Sigma^{\leq n}$ を長さ n 以下の語全体であるとする.

パターンとは定数記号と変数からなる有限文字列である. パターン p の長さを $|p|$ で表す. p の変数を消去して得られる語を $c(p)$, p に含まれる変数の集合を $var(p)$ で表す.

正規パターンとは各変数の出現するが高々1回のパターンである. 正規パターン全体からなる集合を RP で表す.

正規パターン言語 代入とは定数記号をそれ自身に写す, パターンからパターンへの準同形写像である. 代入は変数のパターンによる置換の集合で表現することができる.

(例) $\Sigma = \{a, b, c\}$, $p = axby$, $\theta = \{x := cc, y := xay\}$ とすると, パターン p の 代入 θ 像は $p\theta = accbxay$ となる.

パターン p がパターン q の代入 θ による像であるとき, つまり $p = q\theta$ となるとき, p は q の例化であるといい, また, q は p の汎化であるという. これを $p \preceq q$ で表す.

正規パターン p が生成する言語 $L(p)$ とは変数に長さ0以上の語を代入することによって得られる語の全体である.

$$L(p) = \{w \in \Sigma^* \mid w \preceq p\}$$

正規パターン言語の性質 言語 L_1, L_2 の連接を $L_1L_2 = \{w_1w_2 \mid w_1 \in L_1, w_2 \in L_2\}$ とすると, 正規パターン $aaxbcycdzdd$ の生成する言語は,

$$\{aa\}\Sigma^*\{bc\}\Sigma^*\{cd\}\Sigma^*\{dd\}$$

となる. つまり, 正規パターン言語 $L(aaxbcycdzdd)$ の元となる語は, 接頭語として aa をもち, 重複なく bc, cd がこの順番で出現し, 接尾語として dd をもつ.

このように正規パターン言語は接頭語, ある順番で重複なく出現する語の列, 接尾語等と密接な関連をもつ.

本論文ではパターンとして正規パターンのみを使うので, 混乱の生じない箇所では正規パターンを単にパターンと表記する.

2.2 k-和積パターン言語

ここでは学習対象となる k-和積パターン表現及びその言語を定義する. また, 次章の証明を容易にするために k-和パターン表現, k-積パターン表現と呼ばれる k-和積パターン表現の特別な形をしたものも定義する.

定義 2.1. 和積パターン表現は正規パターンと $\{\wedge, \vee, (,)\}$ からなる有限文字列で以下のように帰納的に定義される.

1. 長さ0の文字列 (λ と表記する) は和積パターン表現である.
2. 正規パターンは和積パターン表現である.
3. P, Q が和積パターン表現ならば $(P \wedge Q)$ も和積パターン表現である.
4. P, Q が和積パターン表現ならば $(P \vee Q)$ も和積パターン表現である.
5. *1-4* で作られるもののみが和積パターン表現である.

和積パターン表現 P に含まれるパターンの集合を $E(P)$ で表す. k-和積パターン表現は和積パターン表現でかつ $\sharp E(P) \leq k$ となるものとする. ただし, $\sharp$ は集合の濃度を表すとする.

k-和積パターン表現の集合を CRP^k で表す. 命題論理と同様に不要な丸括弧 $(,)$ を省いて表現する.

k-和積パターン表現の生成する言語を以下のように定義する. k-和積パターン表現 P がただ1つのパターン p からなる場合は,

$$L(P) = (L(p))$$

また,

$$L(\lambda) = \emptyset$$
$$L((P \wedge Q)) = (L(P) \cap L(Q))$$
$$L((P \vee Q)) = (L(P) \cup L(Q))$$

とする. k-和積パターン言語の族を $CRPL^k$ で表す.

定義 2.2. $p_1,\cdots,p_n$ を正規パターンとする．和パターン表現とは $p_1\vee\cdots\vee p_n$ の形をした和積パターン表現，積パターン表現とは $p_1\wedge\cdots\wedge p_n$ の形をした和積パターン表現である．

また，含まれるパターンの数の上限 k を明記する場合，k-和パターン表現，k-積パターン表現と表記する．

k-和パターン表現の言語は $L(p_1)\cup\cdots\cup L(p_n)$ と表すことができ，k-積パターン表現の言語は $L(p_1)\cap\cdots\cap L(p_n)$ と表すことができる．

3 k-和積パターン言語の包含関係

3.1 k-和積パターン言語の2つの表現

2つの k-和積パターン言語の包含関係にまつわる結果を示すために，まず k-和積パターン言語を2通りに表現する．

和積パターン表現の和積標準形 命題論理における和積標準形への変換と同様に，和積パターン表現を和積標準形に変換することができる．詳細は省略する．和積パターン表現 Q の和積標準形を $NF(Q)$ で表すことにする．$E(Q)=E(NF(Q))$ となることに留意する．

k-和積パターン言語の和による表現 k-和積パターン言語を特定の形をした和パターン言語で表す．

(例)2-和積パターン言語 $L(x_0a_1x_1a_2x_2a_3x_3)\cap L(xcy)$ は，以下のように4-和パターン言語で表される．

$$L(x_0cx_1a_1x_2a_2x_3a_3x_4)\cup$$
$$L(x_0a_1x_1cx_2a_2x_3a_3x_4)\cup$$
$$L(x_0a_1x_1a_2x_2cx_3a_3x_4)\cup$$
$$L(x_0a_1x_1a_2x_2a_3x_3cx_4).$$

この例からも分かるように，k-和パターン言語の族は k-和積パターン言語の族に真に含まれる．

定理 3.1. k-和積パターン言語に等しい和パターン言語が存在する．

Proof. 2-積パターン言語の場合のみを証明すれば十分である．p_1,p_2 を正規パターンとする．$L(p_1\wedge p_2)=\emptyset$ に等しい和パターン言語は $L(\lambda)$ である．従って，空でない2-積パターン言語 $L(p_1\wedge p_2)$ が和パターン言語で表されることを証明する．

$w\in L(p_1\wedge p_2)$ とし，$w=p_1\theta_1=p_2\theta_2$ となる θ_1,θ_2 をそれぞれ1つづつとる．$pt(w,p_j,\theta_j)[i]\in X\cup\Sigma$ $(1\le i\le|w|,j=1,2)$ を次のように定義する．

$$pt(w,p_j,\theta_j)[i]=$$

$$\begin{cases} x_i & w[i]\text{ は }\theta_j\text{ による }p_j\text{ の変数の像}\\ w[i] & w[i]\text{ は }\theta_j\text{ による }p_j\text{ の定数の像}\end{cases}$$

長さ $|w|$ のパターン q_{w,θ_1,θ_2} を

$$q_{w,\theta_1,\theta_2}[i]=$$

$$\begin{cases} x_i & pt(w,p_j,\theta_j)[i]\in X(i=1,2)\\ w[i] & \text{上記以外の場合}\end{cases}$$

q_{w,θ_1,θ_2} の標準形を $\bar{q}_{w,\theta_1,\theta_2}$ とする．$w\preceq\bar{q}_{w,\theta_1,\theta_2}$ であり，$|c(\bar{q}_{w,\theta_1,\theta_2})|\le|c(p_1)|+|c(p_2)|$，$\bar{q}_{w,\theta_1,\theta_2}\le 2|c(\bar{q}_{w,\theta_1,\theta_2})|+1$ が成立する．

従って，異なる $\bar{q}_{w,\theta_1,\theta_2}$ は p_1,p_2 に依存した有限個しかなく，その有限個の正規パターンからなる和パターン表現 Q の生成する言語 $L(Q)$ は $L(p_1\wedge p_2)$ を含む．また，$\bar{q}_{w,\theta_1,\theta_2}$ の構成法より，$L(Q)\subseteq L(p_1\wedge p_2)$ となる．

よって2-積パターン言語に等しい和パターン表現の存在が示された． ■

k-和積パターン表現 P に対し，言語の等しい和パターン表現の集合を $D(P)$ で表す．

3.2 特徴集合と包含関係

ここでは，k-和積パターン言語の特徴集合と包含関係を取り扱う．特徴集合の存在は次章で議論する学習可能性に深く関わる．また，特徴集合の概念を用いて k-和積パターン言語の包含

関係を2つの k-和積パターン表現の構文的関係に還元することも試みる.

語の有限集合 S が言語族 $\mathcal{L}$ における言語 $L \in \mathcal{L}$ の特徴集合であるとは,

$$S \subseteq L' \in \mathcal{L} \Rightarrow L \subseteq L'$$

が成立することである.

2つの k-和パターン表現 P, Q に対して次のような二項関係 $\sqsubseteq$ を定義する.

$$P \sqsubseteq Q \iff \forall p \in E(P), \exists q \in E(Q)\ s.t.\ p \preceq q$$

p を正規パターンとし, $var(p) = \{x_1, \cdots, x_n\}$ する. $S(p)$ を以下のように定義する.

$$S(p) = \{p\{x_1 := w_1, \cdots, x_n := w_n\} \mid w_i \in \Sigma^2\ (1 \leq i \leq n)\}$$

また, 和パターン表現 P に対し,

$$S(P) = \bigcup_{p \in E(P)} S(p)$$

とする.

k-和パターン言語からなる族において閉包性と呼ばれる以下のものが成立する.

補題 3.2 (植村・佐藤 [22]). $\sharp\Sigma \geq k+2$ のとき, P, Q を k-和パターン表現とすると以下の同値性が成立する.

$$S(P) \subseteq L(Q) \iff P \sqsubseteq Q \iff L(P) \subseteq L(Q)$$

補題 3.3. $\sharp\Sigma \geq k+2$ のとき, P を和パターン表現, Q を k-和パターン表現とすると以下の同値性が成立する.

$$\begin{aligned} & (i) S(P) \subseteq L(Q) \\ \iff\ & (ii) P \sqsubseteq Q \\ \iff\ & (iii) L(P) \subseteq L(Q) \end{aligned}$$

Proof. $(ii) \Rightarrow (iii), (iii) \Rightarrow (i)$ は自明である. $(i) \Rightarrow (ii)$ を証明する. $p \in P$ を満たす任意の p に対して, $S(p) \subseteq L(Q)$ より, 上記の補題を用いると, $\{p\} \sqsubseteq Q$ である. これはどの p に対しても成立するから, $P \sqsubseteq Q$ である. ■

k-和積パターン表現 Q に対しその積和標準形を以下のように表現しておく.

$$NF(Q) = I_1(Q) \wedge \cdots \wedge I_m(Q)$$

ただし $I_i = J_{m,1} \vee \cdots \vee J_{m,n_m}\ (1 \leq i \leq m)$ であるとする. $E(I_i) \subseteq E(Q)$ となることに注意せよ.

定理 3.4. $\sharp\Sigma \geq k+2$ のとき, $P, Q \in CRP^k$, $P' \in D(P)$ とすると以下の同値性が成立する.

$$\begin{aligned} & (i) S(P') \subseteq L(Q) \\ \iff\ & (ii) \forall i, P' \sqsubseteq I_i(Q)\ (1 \leq i \leq m) \\ \iff\ & (iii) L(P) \subseteq L(Q) \end{aligned}$$

Proof. $(ii) \Rightarrow (iii), (iii) \Rightarrow (i)$ は自明である. $(i) \Rightarrow (ii)$ のとき, $S(P') \subseteq L(Q)$ より, 任意の $i\ (1 \leq i \leq m)$ に対して, $S(P') \subseteq L(I_i)$ である. 上記補題より, $P' \sqsubseteq I_i$ が成立する. ■

4 k-和積パターン言語の学習

4.1 正例からの帰納学習

帰納的言語の添え字つき族 言語族 $\mathcal{L} = L_1, L_2, \cdots$ が帰納的言語の添字付族であるとは, 次のような帰納的関数 $f : N \times \Sigma^* \to \{0, 1\}$ が存在することをいう.

$$f(i, w) = \begin{cases} 1, & \text{if } w \in L_i \\ 0, & \text{if } w \notin L_i \end{cases}$$

正提示 文字列の無限列 $w_1, w_2, \cdots$ が言語 L の正提示であるとは, $\{w_n \mid n \geq 1\} = L$ が成立することである. 文字列の無限列 $w_1, w_2, \cdots$ の1番目から n 番目までの有限列を $\sigma[n]$ で表し, 初期断片と呼ぶ.

推論アルゴリズム 推論アルゴリズム M とは, 次々に入力を要求し, 次々に出力を生成する実行的手続きのことであり, M の出力を推測と呼ぶ. 文字列の無限列 σ に対して, その初期断片 $\sigma[n]$ が入力された後, M が生成する出力を $M(\sigma[n])$ で表す. 推論アルゴリズム M が入力の列 σ に対して, 添字 $g \in N$ に収束するとは,

ある $m \in N$ が存在し，任意の $n \geq m$ に対して，$M(\sigma[n]) = g$ となることをいう．推論アルゴリズム M が言語 L を正例から極限同定するとは，L の任意の正提示に対して，M が $L = L_i$ なる i に収束することである．また，任意の言語 $L \in \mathcal{L}$ を正例から極限同定する推論アルゴリズムが存在するとき，$\mathcal{L}$ は正例から推論可能であるという．

4.2 k-和積パターン言語の学習

定義 4.1. 語の有限集合 S が言語族 $\mathcal{L}$ における言語 L の有限証拠集合であるとは，$S \subseteq L$ のとき，

$$S \subseteq L' \subset L$$

となる言語 L' が言語族 $\mathcal{L}$ に存在しないことである．

補題 4.2 (Angluin[2]). L を言語族 $\mathcal{L}$ の言語とする．任意の L が有限証拠集合をもち，それを列挙するアルゴリズムをもつことと，言語族 $\mathcal{L}$ が正例から帰納推論可能であることは同値である．

定理 3.1 より，$D(P)$ の元を 1 つ具体的にとる方法が存在する．$P' \in D(P)$ とすると，P' に対し $S(P')$ は 1 つに定まり，その元を列挙することは容易である．

定理 3.4 より，$S(P')$ $(P' \in D(P))$ は $L(P)$ の有限証拠集合となることがわかる．

定理 4.3. $\sharp\Sigma \geq k+2$ のとき，CRP^k は正例から帰納学習可能である．

4.3 パターン言語の和・積・補集合と学習

和積補パターン言語

定義 4.4. 和積補パターン表現は正規パターンと $\{\wedge, \vee, \neg, (,)\}$ からなる有限文字列で以下のように帰納的に定義される．

1. 長さ 0 の文字列 (λ で表す) は和積補パターン表現である．
2. 正規パターンは和積パターン表現である．
3. P, Q が和積補パターン表現ならば $(P \wedge Q)$ も和積補パターン表現である．
4. P, Q が和積補パターン表現ならば $(P \vee Q)$ も和積補パターン表現である．
5. P が和積補パターン表現ならば $\neg(P)$ も和積補パターン表現である．
6. *1-5* で作られるもののみが和積補パターン表現である．

和積補パターン表現 P に含まれるパターンの集合を $E(P)$ で表す．k-和積補パターン表現は和積補パターン表現でかつ $\sharp E(P) \leq k$ となるものとする．

k-和積補パターン表現の生成する言語を以下のように定義する．k-和積補パターン表現 P がただ 1 つのパターン p からなる場合は，

$$L(P) = (L(p))$$

また，

$$L(\lambda) = \emptyset$$

$$L((P \wedge Q)) = (L(P) \cap L(Q))$$

$$L((P \vee Q)) = (L(P) \cup L(Q))$$

$$L(\neg(P)) = (L(P))^c$$

となるものとする．

k-和積補パターン言語が学習不能であること

定理の証明の前に，学習不能を示すための結果を引用する．

言語族 $\mathcal{L}$ の言語 L が集積点であるとは，以下の条件 1, 2, 3 を満たす有限言語の列 $(T_n)_{n \in N}$ が存在することである．

1. $T_1 \subseteq T_2 \subseteq \cdots$
2. $\bigcup_{n=1}^{\infty} T_n = L$
3. $\forall n \in N, \exists L_n \in \mathcal{L}\ s.t.\ T_n \subseteq L_n \subset L$

補題 4.5 (Kapur[6]). 言語 L が $\mathcal{L}$ における有限証拠集合をもつことと，L が集積点でないことは同値である．

$a \in \Sigma$ とする．

$$L = L(xay)$$

$$T_n = L(xay) \cap \Sigma^{\leq n}$$

$$L_n = L(xay) \cap (L(x_0 a x_1 a \cdots a x_{n+1}))^c$$

とおくと, L は和積補パターン言語の族に集積点となることがわかる. 上記結果から L は有限証拠集合をもたず, 和積補パターン言語の族が正例から帰納学習不能であることがわかる.

定理 4.6. k-和積補パターン言語からなる言語の族は正例から帰納学習不能である.

参考文献

[1] D. Angluin, *Finding patterns common to a set of strings*, in: Proceedings of the 11th Annual Symposium on Theory of Computing (1979) 130–141.

[2] D. Angluin, *Inductive inference of formal languages from positive data*, Information and Control **45** (1980) 117–135.

[3] S. Arikawa, T. Shinohara, A. Yamamoto, *Learning elementary formal system*, Theoretical Computer Science **95** (1992) 97–113.

[4] H. Arimura, T. Shinohara, S. Otsuki, *Finding minimal generalizations for unions of pattern languages and its application to inductive inference from positive data*, Lecture Notes in Computer Science **775** (1994) 646–660.

[5] E.M. Gold, *Language identification in the limit*, Information and Control **10** (1967) 447–474.

[6] S. Kapur, *Computational Learning of Languages*, PhD thesis, Technical Report 91-1234, Cornell University.

[7] T. Moriyama, M. Sato, *Properties of language classes with finite elasticity*, IEICE Transactions on Information and Systems **E78-D(5)** (1995) 532–538.

[8] T. Motoki, T. Shinohara, K. Wright, *The correct definition of finite elasticity: Corrigendum to identification of unions*, in: Proceedings of the 4th Annual Workshop on Computational Learning Theory (1991) 375–375.

[9] Y. Mukouchi, S. Arikawa, *Towards a mathematical theory of machine discovery from facts*, Theoretical Computer Science **137** (1995) 53–84.

[10] Y. Mukouchi and M. Sato, *Learning languages generated by elementary formal systems and its application to SH languages*, Lecture Notes in Artificial Intelligence **3244** (2004) 380–394

[11] D. Reidenbach, *Result on inductive inference of extended pattern languages*, Lecture Notes in Artificial Intelligence **2533** (2002) 308–320.

[12] D. Reidenbach, *On the learnability of E-pattern languages over small alphabets*, Lecture Notes in Computer Science **3120** (2004) 140–154.

[13] M. Sato, *Inductive Inference of Formal Languages*, Bulletin of Informatics and Cybernetics **27(1)** (1995) 85–106.

[14] M. Sato, Y. Mukouchi, D. Zheng, *Characteristic sets for unions of regular pattern languages and compactness*, Lecture Notes in Artificial Intelligence **1501** (1998) 220–233.

[15] M. Sato, Y. Mukouchi, *Learning of language generated by patterns from positive examples*, Scientiae Mathematicae Japonicae, **58(2)** (2003) 465–472.

[16] T. Shinohara, *Polynomial time inference of extended regular pattern languages*, Lecture Notes in Computer Science **147** (1982) 115–127.

[17] T. Shinohara, *Inductive inference of formal systems from positive data*, Bulletin of Information and Cybernetics **22** (1986) 9–18.

[18] T. Shinohara, *Inductive inference from positive data is powerful*, in: Proceedings of the 3rd Annual Workshop on Computational Learning Theory (1990) 97-110.

[19] T. Shinohara, *Inductive inference of monotonic formal systems from positive data*, New Generation Computing **8** (1991) 371-384.

[20] T. Shinohara, *Rich classes inferable from positive data*, Information and Computation **108** (1994) 175–186.

[21] R.M. Smullyan, Theory of Formal Systems, Princeton University Press, 1961.

[22] J. Uemura, M. Sato, *Compactness and learning of unions of erasing regular pattern languages*, Lecture Notes in Artificial Intelligence **2533** (2002) 293–307.

[23] K. Wright, *Identification of unions of languages drawn from positive data*, in: Proceedings of the 2nd Annual Workshop on Computational Learning Theory (1989) 328–333.